Remarks

Claims 1-7, 10-12, 15, 17-23, 26-28, 31 and 47-48 are pending. All pending claims stand rejected under 35 USC §103.

Claim Rejections – 35 USC §103: The Examiner rejected Claim 1-7, 10-12, 15, 17-23, 26-28, 31 and 47-48 under §103(a) as being unpatentable over Brown (US Pub. No. 2002/0174180) in view of Pivowar (USPN 6,308,201).

Claim 1 is directed to a coordinated push synchronization method and includes the following combination of elements.

1. detecting changes to a local application data store;
2. identifying a record affected by a detected change;
3. pushing the identified record to a remote application data store;
4. ascertaining whether the pushed record, in its current form as affected by the detected change, has already been replicated or deleted in the remote application data store in order to determine whether the remote application data store will be updated with the pushed record;
5. if not, updating the remote application data store with the pushed record and identifying the pushed record in the remote application data store as having been pushed from the local application data store to the remote application data store, otherwise ignoring the pushed record.

With respect to fourth element listed above, the Examiner asserts that Brown, paragraphs [0071 and 0080]-[0083], teaches "ascertaining whether the [pushed] record, in its current form as affected by the detected change, has already been replicated or deleted in the remote application data store in order to determine whether the remote application data store will be updated with the pushed record; if not, updating the remote application data store with the pushed record." The

Examiner Admits that Brown teaches that this act is performed before any data is pushed. To remedy Brown's deficiency, the Examiner mistakenly relies on Pivowar. The Examiner states that Pivowar, col. 11, lines 15-35 and Fig. 15, "discloses wherein the record is pushed to the remote application data store prior to determining whether the record should be updated."

Pivowar is directed to a system utilizing a server (104) to synchronize data between personal digital assistants (PDAs102). *See, e.g.* Pivowar, Abstract and Fig. 1. Between each PDA and server (104) lies a computer (106) that controls, via a client messenger (132), the data that is retrieved from the server (104) and stored on a given PDA (102). *See* Pivowar, col. 11, line 16 through col. 12, line 15 and Figs. 15-16.

The particular passage from Pivowar relied upon by the Examiner is reproduced as follows:

> With reference now to FIG. 15, illustrated is a detailed process of the client messenger 132 that is executed during operations 906 and 1004 of the client messenger 130 shown in FIGS. 9 and 10, respectively. The process of FIG. 15 begins by determining whether a connection with the server 104 exists in decision 1100. As mentioned earlier, such connection may be made by any means including the Internet. Once it is ascertained that a connection exists, a record list, or SyncRecordList, is started in a manner that will be set forth later in greater detail. See operation 1102 of FIG. 15. Once the record list is started, a loop begins with the retrieval of a next local record in operation 1004. As long as the record is not null as determined in decision 1106, a determination is made whether the record represents a change from a previous synchronization in operation 1108 and, if so, is added to the record list in operation 1110.
>
> As shown in FIG. 15, if it is determined that the current record is null in decision 1106, the query is sent to the server 104 in operation 1112 after which the client messenger 132 waits for an update in operation 1114. Finally, a process update in operation 1116 is carried out in a manner to be set forth later in greater detail.

Pivowar, col. 11, lines 16-35.

The Examiner mistakenly contends that this passage discloses pushing a record to a remote application data store prior to determining whether that record

should be updated. This is simply not true. The passage mentions nothing of "pushing" a record. To help illustrate, Figs. 1 and 15 are reproduced below.

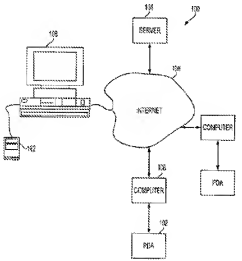

FIG. 1

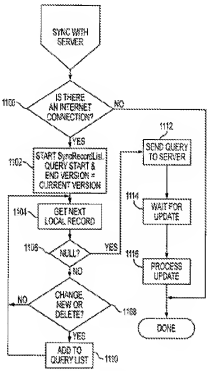

FIG. 15

The passage describes an interaction between a server (104) and client messenger (132) operating on a computer (106). In steps 1104 and 1108, the client messenger gets a local record and determines if that local record is new, has been deleted, or has been changed. If so, that local record is added to a query list in step 1110 that is sent to the server (104) in step 112. The client messenger (132) then processes any updates it receives in step 1116.

In the context of Claim 1 and the cited passage, Pivowar's computer (106) and PDA (102) represent a local application data store. Pivowar's server (104) represents a remote application data store. The passage cited by the Examiner discusses identifying local records that are new, have been changed, or have been

deleted and adding those records to a query list. Pivowar, col. 11, lines 27-31. Those records are pushed to the server (104). *See* Pivowar, step 1112. Pivowar mentions nothing of the server (104) making any determination as to whether or not the server should be updates with a corresponding record pushed that has already been pushed by the client messenger (132).

Furthermore, the passage mentions nothing of Pivowar's server (104) "pushing" a record to the computer (105) / PDA (102). To the contrary, Pivowar describes that its client messenger (132) sends a query to the server (104) and then processes responses. *See* Pivowar, steps 1112 and 1116 in Fig. 15. Because the client messenger (132) makes the initial query, any responses are "pulled" by Pivowar's client messenger (132). They are NOT pushed.

Consequently, Brown even when modified with the teachings of Pivowar, does not teach or suggest ascertaining whether a pushed record, in its current form as affected by a detected change, has already been replicated or deleted in a remote application data store in order to determine whether the remote application data store will be updated with the pushed record.

With respect to fifth element of Claim 1 listed above, the Examiner asserts that Brown teaches "identifying the [pushed] record in the remote application data store as a pushed record (paragraph 0066) and identifying the [pushed] record in the remote application data store as having been pushed from the local application data store to the remote application data store, otherwise ignoring the [pushed] record (paragraph 0071)."

Brown, paragraph [0066], is reproduced as follows:

> [0066] Metadata 330 shows the metadata for file 320 as stored within CSFS 335, part of client SA 337. (Although metadata are not shown for the other files and folders within folder 302, a person skilled in the art will recognize that such metadata exist.) In metadata 330, file 320 is shown as having a name (which is typically not encrypted, although the name can be encrypted in an alternative embodiment of the invention), a CID of 0x62, a SID of 0x2A, a FSI of 35, the change time of the file, and the flags used in the synchronization process (**such as identifying metadata items that**

> **need to be pushed to the server**). Note that metadata 330 is not shown to store the data of file 320, which is stored in the native operating system of computer 130 within the folder structure, as expected.

Brown, paragraph [0066] (emphasis added). Brown's paragraph 66 mentions nothing of identifying a record in the remote application data store as a pushed record. It simply discusses a flag that identifies metadata (not a record) that needs to be pushed to a server.

Brown, paragraph [0071], is reproduced as follows:

> [0071] The client polls the server for changes by other clients by passing its current CSI to the server in a sync polling call. If the CSI matches the server account's SSI value, then the client is up to date with the server. Otherwise the client SA requests server synchronization data (SSD). The SSD contains the following data:

Brown, paragraph [0071]. Brown's paragraph 71 mentions nothing of identifying a record in the remote application data store as having been pushed from the local application data store to the remote application data store

Consequently, Brown and Pivowar fail to teach a method that includes identifying the pushed record in the remote application data store as having been pushed from the local application data store to the remote application data store as recited by Claim 1.

For at least these reasons, Claim 1 is patentable over Brown and Pivowar. Claims 2-4 and 47 are thus also felt to distinguish over those references based on their dependency from Claim 1.

Claim 5 is directed to a coordinated user-initiated synchronization method and includes the following combination of elements:

1. detecting changes to a local application data store;

2. identifying a record affected by a detected change;
3. ascertaining whether the identified record, in its current form as affected by the detected change, was pushed to the local application data store from a remote application data store; and
4. if not, synchronizing the remote application data store with the local application data store.

With respect to the third element listed above, the Examiner admits that Brown fails to teach or suggest "ascertaining whether the identified record, in its current form as affected by the detected change, was pushed to the local application data store; and if not, synchronizing the remote application data store with the local application data store." Initially it is noted that the Examiner is mischaracterizing that which Claim 5 recites – specifically – "ascertaining whether the identified record, in its current form as affected by the detected change, was pushed to the local application data store **from a remote application data store**" (emphasis added). The Examiner's discussion of Claim 5 ignores this emphasized portion of Claim 5. As such, the rejection is improper.

Nonetheless, the Examiner mistakenly relies on Pivowar to remedy Brown's deficiency. Specifically, the Examiner asserts that Pivowar, col. 9, line 55 through col. 10, line 35, teaches "ascertaining whether the identified record, in its current form as affected by the detected change, was pushed to the local application data store; and if not, synchronizing the remote application data store with the local application data store." The cited passage references Pivowar's Fig. 12. The passage and figure are reproduced below to illustrate the Examiner's mistake.

> FIG. 11 illustrates the various server data that is stored on the server 104. Such data includes a plurality of record lists, or SyncRecordLists 700, that are communicated with the PDA's 102 via the conduit 128 of FIGS. 3 and 4. The record lists include information such as a type number, an identification number, a start version number, an end version number, etc. Further, the record lists contain records, or SyncRecords 702, each including a local identification code; a shared identification code identifying users with whom data is shared; flags to indicate new, deleted, and conflicted information; etc. In

addition, each of the records includes any information that has changed since the last synchronization.

With reference to FIG. 12, a flowchart is provided which delineates the process associated with the server 104 of the present invention. As shown, the process begins with operation 800 wherein a specific record list, SyncRecordList, of records, or SyncRecords, is found or created in response to a query made by a conduit 128. A loop is then executed during which the records are retrieved in operation 802 and then monitored in decision 804 to determine whether records have changed since the last synchronization. If changes have occurred on the record, such record is added to the record list as indicated in operation 806. It should be noted that only changed information is included in the added record. The loop continues until all of the records have been checked for changes in decision 808.

With continuing reference to FIG. 12, the process associated with the server 104 continues with operation 810 wherein a query record is retrieved. If the query record is resident in a response list as indicated in decision 812, the record is marked as conflicted and is dealt with in operation 813. If, however, the query record is not in the response list, it is next determined what type of change has occurred in decision 814.

If a "new" change has occurred, a new record is made and the version number of the record list is incremented. See operation 816 in FIG. 12. It should be noted that this incremented version number is used as an identification code for the new record. If a "modified" change has occurred, a new record is added to the record list and the version number is incremented, as indicated in operation 818. Finally, operation 820 is carried out if a "delete" change has occurred. In such case, a new record is added to the record list, the version number is incremented, and the record is marked as deleted. The forgoing process is continued until no more records exist. Finally, the update is returned in operation 822.

Pivowar, col. 9, line 55 through col. 10, line 35.

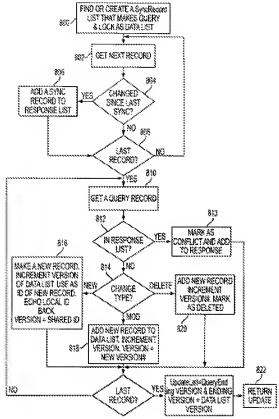

FIG. 12

The cited passage and Fig. 12 relate to actions taken be Pivowar's server (104) discussed above. In short, steps 802 through 808 involve the server (104) identifying which of its records have changed and adding those records to a response list. Steps 810-813 involve the server (104) identifying records queried by the client messenger (132) that are on the response list. Such records are put through a conflict process. Steps 814-820 involve the Server (104) adding all other query records to the response list to be returned as an update in step 822.

The cited passage from Pivowar mentions nothing of ascertaining whether the identified record, in its current form as affected by the detected change, was pushed to the local application data store **from a remote application data store**; and, if not, synchronizing the remote application data store with the local application data store.

Pivowar makes no mention of ascertaining the source of a new or updated record. Consequently, Pivowar does not teach or suggest ascertaining whether a given record was pushed to the local application data store from any particular location let alone pushed from a remote application data store.

For at least these reasons, Claim 5 is felt to distinguish over Brown. Claims 6, 7, and 48 are thus also felt to distinguish over Brown based on their dependency from Claim 5.

Claim 10 is directed to a coordinated push and user-initiated synchronization method and includes the following combination of elements.

1. detecting changes to a local application data store;
2. identifying a first record in the local application data store affected by a detected change;
3. pushing the first record to a remote application data store;
4. ascertaining whether the pushed record, in its current form as affected by the detected change, has already been replicated in or deleted from the remote application data store and, if not, updating the remote application data store with the pushed record;
5. detecting changes to the remote application data store;
6. identifying a second record in the remote application data store affected by a detected change;
7. ascertaining whether the second record, in its current form as affected by the detected change, has already been pushed into the remote application data store in order to determine whether the remote application data store will be updated with the pushed record and, if not, synchronizing the remote application data store with the local application data store, otherwise ignoring the pushed record.

As with Claim 1, Brown and Pivowar fail to teach a method that includes "ascertaining whether the pushed record, in its current form as affected by the detected change, has already been replicated in or deleted from the remote application data store and, if not, updating the remote application data store with the pushed record"

For at least these reasons, Claim 10 is felt to distinguish over Brown. Claims 11, 12, and 15 are thus also felt to distinguish over Brown based on their dependency from Claim 10.

Claim 17 is directed to a computer readable medium having instructions for performing the method steps of Claim 1. For the same reasons Claim 1 distinguishes over Brown so does Claim 17. Claims 18-20 are thus also felt to distinguish over Brown based on their dependency from Claim 17.

Claim 21 is directed to a computer readable medium having instructions for performing the method steps of Claim 5. For the same reasons Claim 5 distinguishes over Brown so does Claim 21. Claims 22 and 23 are thus also felt to distinguish over Brown based on their dependency from Claim 21.

Claim 26 is directed to a computer readable medium having instructions for performing the method steps of Claim 10. For the same reasons Claim 10 distinguishes over Brown so does Claim 26. Claims 27, 28, and 31 are thus also felt to distinguish over Brown based on their dependency from Claim 26.

Conclusion: All pending claims are felt to be in condition for allowance. The foregoing is believed to be a complete response to the outstanding office action.

Respectfully submitted,
Richard Detweiler, et al

By /Jack H. McKinney/

Jack H. McKinney
Reg. No. 45,685

May 15, 2006